--RELATED U.S. APPLICATION

A1 This application is a Continuation of the copending patent application, US Patent No. 6,421,059, with issue date July 16, 2002, assigned to the assignee of the present application and hereby incorporated by reference. --

The paragraphs starting on page 6, line 19, and ending on page 7, line 19, are amended as follows:

-- Figure 3 illustrates a glyph or a compressed font 300 produced in accordance with the second conventional technique. Pixels are represented by a bit of information (i.e., a 1 or a 0). As is seen, each bit 302 represents one of the

A2 bytes of the information that was shown previously in Figure 2. Accordingly, rather than having to provide 35 bytes of information in the case of an 8 bit frame buffer, in this environment the CPU 12 (Figure 1) would only need to provide 35 bits across the bus 16, which translates into 4 3/8 bytes of information. These 4 3/8 bytes of information therefore can be decompressed utilizing circuitry in the graphics controller 24.

Figure 4 illustrates a graphics controller 24 and a frame buffer 26 in accordance with the second conventional technique. Typically, the graphics controller 24 will include an expansion block 302 which will render the pixel into the frame buffer 26 based upon the glyphs. The expansion block 302 includes first and second registers 312 and 314. A multiplexer 310 within the expansion block 302 is utilized to select between the two registers 312 and 314 based upon the glyph. The selection of the registers 312 and 314 is based upon the bits provided by the CPU

12 (Figure 1). Accordingly, if the bit provided by the CPU 12 is a foreground color, the multiplexer 310 selects the register 312 which would expand the bit to a plurality of bits (i.e., 8 bits, 16 bits or 32 bits) and those bits are then provided into the frame buffer 26 as a foreground color. On the other hand, if the bit provided is background color, the multiplexer 310 selects register 314 which would expand the bit into a plurality of bits and those bits are then provided to the frame buffer 26 as a background color. As further improvement in the second embodiment, a so-called transparent mode can be provided. In this mode, a multiplexer 311 allows for the background color to be provided from the frame buffer 26 via select line 315 and input 313. For example, if the background color is 00 then the frame buffer automatically loads zeros for the background color of that particular character. In so doing, rendering time is further reduced. Accordingly the character 100 that is rendered in the frame buffer is exactly the same as that shown in Figure 1. --

The paragraph on page 9, starting at line 4, is amended as follows:

-- To discuss the present invention in the context of a preferred embodiment, refer now to Figure 5 and the accompanying discussion. Figure 5 is a block diagram of a graphics controller 400 and a frame buffer 450 in accordance with the present invention. The graphics controller includes an expansion unit 402 that has similar components to that of expansion unit 302 of Figure 4. The memory 450 includes a data structure 451. The data structure 451 includes a plurality of font arrays 460 and 462. It should be understood that there could be any number of font arrays in the data structure 451. As is seen, in font array 462 the pitch of the font characters is larger than the pitch of the font characters in font array 460. Hence, for each font array 460 and 462 a different index and a

different font pointer are required to allow the graphics controller to access the associated font characters. Accordingly, a font pointer is changed when, for example, the font is changed. Each of the font arrays 460 and 462 include information concerning the size of each character and an index that indicates the location of each font character within the font array. The font characters can be either full sized fonts or glyphs. The information within the font array is utilized by the graphics controller 400 to allow the graphics controller 400 to render a particular character retrieved from the memory 450. --

A3 cont.

Line 19 on page 10 is amended as follows:

A4

-- SIZE Width Register 420 --

The paragraph on page 12, beginning at line 9, is amended as follows:

-- In an example, all the user needs is to load the index value, the x value and the y value for the character that is to be rendered, where the index is an ASCII character number. In a preferred embodiment, the x value can be 12 bits, the y value can be 12 bits and the index value can be 8 bits. In so doing, one 32 bit transfer from the CPU to the graphics controller is all that is needed to render the character. --

A5

IN THE CLAIMS

The claims are amended as follows.